express mail label no. EL875326485US 80-00

APPARATUS, METHODS AND ARTICLES OF MANUFACTURE FOR INTERCEPTING, EXAMINING AND CONTROLLING CODE, DATA AND FILES AND THEIR TRANSFER

FIELD OF THE INVENTION

The present invention relates to apparatus, methods and articles of manufacture for intercepting, examining and controlling code, data and files and their transfer. More particularly, the present invention relates to apparatus, methods and articles of manufacture for intercepting, examining and controlling proscribed or predetermined code, data and files and their transfers.

BACKGROUND OF THE INVENTION

The rise of the Internet and networking technologies has resulted in the widespread transfer of code, data and files between computers. This material is not always what it seems to be. For example, code that is accessed on a remote machine and downloaded to a computer system can contain hostile algorithms that can potentially destroy code, crash the system, corrupt code or worse. Some of these hostile algorithms are viruses, worms, and Trojan horses.

Hostile, malicious and/or proscribed code, data and files ("code" as used hereinafter generally includes "data" and "files") can infect a single computer system or entire network and so posit a security risk to the computer system or network. The user and/or administrator (generally referred to hereinafter as "user") may wish to intercept, examine and/or control such code. The user might also wish to intercept, examine and/or control other code as well, for example, code which the user does not know to be hostile, but wishes to intercept nonetheless, for example, potentially sexually or racially harassing email, junk email, trade secret text, or other confidential information, etc. This latter type of code is known hereinafter as "predetermined code".

Antivirus or other similar packages attempt to protect the system or network from hostile, malicious, predetermined and/or proscribed code (generally referred to hereinafter as "proscribed code.") VFIND®, from CyberSoft, Inc., is one such product that may protect systems and networks from proscribed code. If the virus programs are not run frequently -- an all too common occurrence -- they will not protect the system. Therefore, the benefits and protections offered by antivirus programs are often lost.

The difficulty of scanning code or proscribed code is accentuated by email. Email, providing a simple and convenient method of transferring code, is often only scanned after receipt, at the user's option. If the user does not scan the email, or improperly scans the email, proscribed code might infect the system. Moreover, programs often used to send and receive email, such as Microsoft Outlook®, may open the email automatically and thus permit proscribed code to infect the system without any user interaction whatsoever. In such a situation, the user may not even realize his or her system is infected until too late – after the infection by the proscribed code.

Moreover, a primary method of detecting viruses and other hostile code is by examining the code only after it has entered the user's machine. This method may provide some protection however the virus may still be on the user's machine and available to the network.

Therefore, it would be beneficial to have apparatus, methods and articles of manufacture for simply and effectively scanning email in an efficient manner transparently or almost transparently to the end-user, with little or no operational effort required by the user.

Accordingly, it is an object of the present invention to provide apparatus, methods and articles of manufacture that simply and effectively intercept, control, and/or examine incoming and outgoing code in an efficient manner transparently or almost transparently to the end-user, with little or no operational effort required by the user.

It is a further object of the present invention to provide apparatus, methods and articles of manufacture that simply and effectively intercept, control, and/or examine incoming and outgoing code transferred, at least in part, through a "store and forward" transfer system, in an efficient manner transparently or almost transparently to the end-user, with little or no operational effort required by the user.

It is a further object of the present invention to provide apparatus, methods and articles of manufacture that simply and effectively intercept, control, and/or examine incoming and outgoing code transferred, at least in part, through a "store and forward" transfer system, in an efficient manner transparently or almost transparently to the end-user, with little or no operational effort required by the user.

SUMMARY OF THE INVENTION

The present invention comprises apparatus, methods and articles of manufacture for intercepting, examining, and/or controlling code transferred, at least in part, through a "store and forward" system (hereinafter "stored and forwarded code.") The present invention may operate on a single computer system, network, or multiple systems or networks as desired.

The present invention may, in various embodiments, process, that is, intercept, examine, and/or control, any or all stored and forwarded code in a computer or network. Intercepting, examining and/or controlling stored and forwarded code includes but is not limited to sorting, altering, monitoring, blocking, logging, quarantining, discarding, redirecting and/or transferring code. Although the present invention can be implemented on various platforms, the preferred embodiments are used in Unix® and various Windows® environments, such as NT, 2000, 95, 98 and Me.

The especially preferred embodiments of the present invention process stored and forwarded email. Email is usually stored and forwarded through a queue, and the especially preferred embodiments create a new, secondary queue prior to further sendmail processing. A transfer component retrieves the messages from the queue and delivers them in turn to a proscribed code scanner prior to further sendmail processing. For example, in Unix® environments using sendmail, the preferred embodiments will create at least one new, secondary queue and transfer messages to that secondary queue by way of a transfer component, as the messages are scanned for proscribed code using a proscribed code scanner. If any particular message contains proscribed code, the message could be altered, blocked, logged, etc. In the preferred embodiments, the messages containing proscribed code are placed in another new, secondary queue, and the user or administrator could be notified, or not, as desired.

Additionally, preferred embodiments may have more than one level of transfer component or be "multilevel." In such a multilevel embodiment a primary transfer component transfers messages to second level transfer components based on one or more parameters such as size of the message, etc.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 is a schematic diagram of the operation of sendmail.

Figure 2 is a schematic diagram of a preferred embodiment of the present invention.

Figure 3 is a schematic diagram of another preferred embodiment of the present invention.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

The present invention comprises apparatus, methods and articles of manufacture for intercepting, examining, and/or controlling code transferred, at least in part, through a "store and forward" system. In a stored and forwarded system, code is stored or queued (also referred to herein as "intermediate storage") at some point along the transmission, and then forwarded to the recipient. A stored and forwarded system may maintain its intermediate storage in a number of ways or components. For example, storage components may be located in memory, on disk, on another system, etc. Storage or queuing is used for a number of reasons: for example, if the transmission pathway is blocked or the destination is unreachable, a queue may maintain the messages for some period of time, in order to try transmitting the message again.

The preferred embodiments process, that is, intercept, examine, and/or control, stored and forwarded code, including email, other message code, and other stored and forwarded code. "Stored and forwarded code" is defined herein as discrete units of code, stored and forwarded as those discrete units.

The stored and forwarded code processed by the embodiments of the present invention may be transferred through any number of connections in a computer system, systems, network or networks. Processing code, that is, intercepting, examining and/or controlling code, includes but is not limited to sorting, altering, monitoring, blocking, logging, quarantining, discarding, redirecting and/or transferring email.

An especially preferred embodiment of the present invention runs on a Unix® platform with sendmail such as System V, Sun Solaris®, IBM AIX®, HP-UX®, etc. The following description of the preferred embodiments uses Sun Solaris® operating system Unix® terminology. However, it should be specifically understood that embodiments can be implemented in other Unix® and Unix®-like platforms, including but not limited to Linux® and its variants, as well as other operating system platforms including but not limited to Microsoft Windows® NT, Windows® 2000, Windows® 95, 98 and Me, IBM MVS, IBM OS/390, SNA, 3270 RJE, MacOS, VxWorks® and others. Moreover, embodiments of the present invention may be used in cross platform situations, such as for example, in a network using SMTP to transfer messages, or for example, in an enterprise running IBM's MQSeries of products which provides, inter alia, enterprise-wide messaging capabilities using store and forward technology.

The preferred embodiments are written in UNIX Bourne shell script, with components written in other languages, although any language known in the art may be used.

In typical email technology, the user has a mail user interface or mail user agent (MUA) to compose, read and send email. The MUA transmits the email from the user to a mail transport agent (MTA.) The MTA then makes routing and delivery decisions, transmits the email between machines, etc.

Sendmail is a common MTA in UNIX environments. Before turning to the especially preferred embodiments operating with sendmail, it would be helpful to review the operation of sendmail. Sendmail is a group of programs, files, directories and services installed on a user's mail processing machine. (Typically, most UNIX machines are connected in a network, and one of the networked machines functions as a mail processing machine for the network users.)

In typical operation, sendmail receives email from another source or sources and passes it to the user or users. One of the components used by sendmail to accomplish this function is a queue which holds mail until it can be delivered. The queue is a directory, usually on the mail processing machine. The queue stores outgoing messages, i.e., those messages to be sent to other users; as well as incoming messages, i.e., those messages sent from other users.

A message stored in the queue is comprised of two primary parts: a message header containing the address and other "envelope" or routing and delivery information; as well as a message body, or the actual message material. The message header and message body are stored in separate files in the queue directory. Additionally, other related files may be stored in the queue directory, such as lock files which are used to insure message integrity. The queue directory is usually called **mqueue**. Sendmail's Queue directory addresses and other parameters are modified through various configuration variables invoked by command line options, or by changes to a sendmail configuration file.

Once initiated, sendmail usually resides as a daemon on the system, listening on the appropriate connection (usually port 25) for message transmission. When an incoming message is detected, sendmail will fork one or more children, which will then store and forward the email.

Turning now to Figure 1, an example of a sendmail process is seen. Sendmail 1 receives the incoming message, such as Message D, and separates the messages it receives into message headers and message bodies, such as Message C Header and Message C Body. These are stored in the queue directory in two files, called in this example **qf** and **df,** respectively. Sendmail then forks a sendmail child process, Sendmail 2, then initiates a TCP/IP connection to the next destination for the message (which may be another user's MUA, another MTA, etc.,) ensures the recipient's address exists, removes the message from the queue, reassembles the message and delivers the message.

The preferred embodiments implement proscribed code scanning of the messages stored in a queue. Turning now to Figure 2, a schematic diagram of the especially preferred embodiment is shown. In this embodiment, a single machine is serving as the mail hub. The machine has sendmail installed and the sendmail queue has been created. The embodiment comprises a transfer component, a proscribed code scanner, and four secondary storage components, or queue directories.

It is important to note that the number of secondary storage components or queue directories used in any particular embodiment is as desired: for example an embodiment might comprise a transfer component, a proscribed code scanner, and one secondary queue directory. It might usually be advantageous to use a secondary queue for secondary storage components that is the same type as a first storage component, for example, in a sendmail embodiment it would usually be advantageous to construct a secondary sendmail queue because a sendmail delivery process could be fairly easily configured to pick up mail from that secondary queue. However, it should be noted that embodiments may use any type or number of secondary storage components, or dispense with a secondary stored component entirely. For example, preferred embodiments might use some type or number of secondary storage component or components, use no secondary stored component by transferring code directly to a subsequent messaging or other application, etc.

In the preferred embodiments, the secondary queue or queues used may be created upon installation or startup, as desired. Sendmail 2a has also been modified to point to Queue 2a for outgoing messages rather than the original Queue 1. The specific port is as desired, however, care should be taken to insure that the port chosen is not being used by other applications. This modification in this embodiment was accomplished by way of a command line, although other methods of modification are possible such as to a sendmail configuration file, etc.

In this embodiment, sendmail forks a child process Sendmail 1a when it detects an incoming message. Sendmail 1a parses the message into header and body, and those header and body components are stored in Queue 1a. Copies of the header and body components are then made by Transfer Component 1a, reassembled into the message and passed to Proscribed Code Scanner.

In some embodiments, code information, e.g. location information, directory information, etc., rather than or in addition to code or copies of code might be transferred. The transfer of this information allows for subsequent operations in these embodiments, e.g., proscribed code scanning, etc. Therefore the word "transfer" as used herein with regard to code or messages is intended to encompass transfer of code, copies of code and code information, any and/or all of which can be used in the various embodiments of the present invention.

After Proscribed Code Scanner scans the message for proscribed code, it returns an indicator of the result of the scan to Transfer Component 1a. This proscribed code indicator may take many forms: e.g. whether the content is acceptable, that is, has no proscribed code; whether the message is virus infected; whether the message is merely spam, etc. Transfer Component 1a moves the header and body components to the appropriate queues, (Queue 2a, Queue 3a, Queue 4a or Queue 5a) based on the indication from the Proscribed Code Scanner as described above.

In especially preferred embodiments, a proscribed code scanner and transfer component are able to communicate in order to assist the process. For example, a transfer component might well use the same or similar flags or other indicators of a proscribed code scanner if the proscribed code scanner is a self-contained engine, such as VFIND® by CyberSoft, Inc. This type of information exchange would be also helpful in a number of other ways, for example, to interrogate a proscribed code scanner in order to understand the scanner's messaging processing status, etc.

Returning now to the embodiment of Figure 2, each secondary queue contains a different category of messages or attachments after processing by proscribed code scanner: secondary queue directory Queue 2a contains messages that have passed the scanning and may now be processed by Sendmail 2a accordingly; secondary queue directory Queue 3a contains messages that are infected by a virus; secondary queue directory Queue 4a contains messages that qualify as junk mail or spam; and, secondary directory Queue 5a contains messages that contain confidential material that is not to be sent by email. In other embodiments there may be more or fewer secondary queue directories, as desired, containing any sort of code categories. For example, one embodiment of the present invention may sort mail, or other stored and forwarded code, into categories, for example by size. The number of secondary queue directories in this type of embodiment could then depend upon message sizes, with different sizes being placed into different secondary queues. Such an arrangement would assist in preventing message lag, wherein large messages would take more time to pass through the system and so block smaller messages. By placing larger messages into a secondary queue or queues separate from the secondary queues of smaller messages, the smaller messages could proceed through the system more quickly.

In some preferred embodiments, the message header provides information to be used for decisions by a transfer component. For example, an embodiment may implement a number of proscribed code scanners, each with different settings for scanning different code. Messages may be sent to a particular scanner by a transfer component according to header information, i.e., a previously untrustworthy header might sent to a virus proscribed code scanner, etc. Of course a header indicating spam might be sent directly to a queue in certain embodiments, without going through a proscribed code scanner first.

Of course, as discussed above, other embodiments may use other arrangements and other numbers of secondary queues as desired. As an example, if a store and forward process uses more than one original queue, more than one secondary queue may be created.

Returning now to the embodiment of Figure 2, once the messages are stored in the secondary queues, those in Queue 2a will be processed by Sendmail 2a for subsequent delivery. The messages stored in the other secondary queues may be disposed of, modified, stripped of offending material, etc. or otherwise treated in any manner as desired. For example, the infected messages and/or attachments may be brought to the user, administrators, or another's attention. As should be clear, any type of stored and forwarded code may be intercepted, examined, and/or controlled according to the embodiments of the present invention. In some embodiments, for example, the proscribed code scanner may be reviewing the code for sexually or racially harassing material, for corporate trade secrets, or for any other predetermined code. Additionally, in various embodiments, the transfer component may itself classify code according to various parameters as mentioned above.

Turning now to Figure 3 another preferred embodiment, one with numerous transfer components, is seen. In this embodiment there are a number of transfer components: Transfer Component 1b or a primary transfer component; and secondary transfer components, 1c, 1d, 1e, 1f and 1g. This embodiment, and other multiple transfer component embodiments which generally use one or more primary transfer components to feed one or more other secondary and possibly other level transfer components, would be especially useful in a number of circumstances. For example, multiple transfer component embodiments might be used for load distribution, resource and/or processor management in single or multiple processor system, systems, network or networks.

Transfer Component 1b has no associated proscribed code scanner. Rather, Transfer Component 1b scans the messages in Queue 1b and delivers them to various secondary queues according to size. This process helps insure that larger messages are reviewed appropriately while permitting smaller messages to proceed around the larger messages thus minimizing chances of a stalled system or process. In this embodiment, Queue 2b receives the largest messages, Queue 6b the smallest, and the remaining Queues take various other sizes. The exact size demarcations are as desired, and may be dependent on any of a number of factors such as type of system in which the embodiment is installed, type of messages passing through the embodiment, etc. Other embodiments might deliver messages according to other parameters such as message lag time (length of time message has been in the system,) etc.

Returning to Figure 3, messages are sent by Transfer Component 1b to the appropriate size differentiated queue. The secondary Transfer Components associated with the queue then reviews the code for proscribed code by way of a proscribed code scanner, in a process like that described above with regard to Figure 2. In the embodiment of Figure 3 and other preferred embodiments, each Transfer Component has an associated Proscribed Code Scanner. In other embodiments, there may be a different ratio of Proscribed Code Scanners to Transfer Components.

The message is then routed appropriately, according to the outcome of the proscribed code scan, into an appropriate Queue for final disposition. For example, in the embodiment shown in Figure 3, mail that has passed the scan is sent to Queue 2b, for routing and delivery by Sendmail 2b.

It should be noted that, in the various embodiments of the present invention, stored and forwarded code may be routed, or not, as desired, from a secondary storage component. For example, in the embodiment of Figure 3 dotted lines show various possible destination for the code retained in the various secondary queues. For example, Destination A could be a storage area on the administrator's machine, Destination B a storage area on a file server, Destination C a storage area on an antivirus manufacturer's network, etc. Additionally, monitoring and/or communication components might be used in various embodiments, such as, for example, monitoring the status of transfer components, message flow through the system, the number of virus files, communication of status or other information between components, etc. Any monitoring components added to various embodiments may be added to a number of components, such as a transfer components, a proscribed code scanner, a secondary storage component, etc. and may include logging and/or other reporting components, such as notification components.

In some embodiments, code transfer might be on any batch or other basis, such as through a specific number of messages on a regular cycle, etc. For example, some specific number of messages, such as 20, might be processed at regular intervals. In other embodiments, stream processing might occur. For example, one especially preferred embodiment passes messages from a transfer component to a proscribed code scanner, and, as the transfer component receives proscribed code indications from the scanner, the component passes the messages to a secondary queue for immediate delivery by a sendmail or other mail process.

In some embodiments, a secondary storage component need not be present. For example, embodiments may transfer code directly to a sendmail process or other transfer agent or component. These embodiments may use known API's or other EDI's as known in the art.

In alternate embodiments, the invention comprises an article of manufacture, or signal-bearing medium, containing computer readable code. Examples of such articles include tarred code and other types and/or methods of storing, archiving and/or compressing code known in the art, contained on any media known in the art, such as CD-ROM's, floppy disks, etc.

The above description and the views and material depicted by the figures are for purposes of illustration only and are not intended to be, and should not be construed as, limitations on the invention. Moreover, certain modifications or alternatives may suggest themselves to those skilled in the art upon reading of this specification, all of which are intended to be within the spirit and scope of the present invention as defined in the attached claims.